Grafting Locomotive Motions

Shrinath Shanbhag
Indian Institute of Technology Bombay
India 400076
svs@it.iitb.ac.in

Sharat Chandran
Indian Institute of Technology Bombay
India 400076
sharat@cse.iitb.ac.in

ABSTRACT

The notion of transplanting limbs to enhance a motion capture database is appealing and has been recently introduced [Sha04], [Ike04]. A key difficulty in the process is identifying believable combinations. Not all transplantations are successful; we also need to identify appropriate frames in the different clips that are "cut-pasted." In this paper, we describe motion grafting, a method to synthesize new believable motion using existing motion captured data. In our deterministic scheme designed for locomotive actions, motion grafts increase the number of combinations by mixing independent kinematics chains with a base motion in a given clip.

Our scheme uses a cluster graph data structure to establish correlation among grafts so that the result is believable and synchronized.

Keywords
Motion capture, motion synthesis, cluster graph, motion grafting

1. INTRODUCTION

Movies and interactive applications such as games use virtual humanoid actors extensively. These virtual actors are modeled using hierarchical skeletons consisting of 30 or more degrees of freedom. Animating such characters is a daunting task at best. The animation data for such characters can be generated in one of three ways: (i) it can be produced procedurally by simulating various physical processes, (ii) it can be handcrafted painstakingly by skilled animators using forward/inverse kinematics based systems or (iii) it can be acquired directly from a live performer using motion capture devices. Method (iii) has gained wide acceptance in recent times because motion capture is the fastest way to generate rich, realistic animation data. In addition motion capture (mocap) techniques can capture even individual nuances of a performer and thus produce very realistic animation.

Conference proceedings ISBN 80-86943-03-8
WSCG'2006, January 30-February 3, 2006
Plzen, Czech Republic.

Having said, this, we recognize that human beings are active entities that produce innumerable actions. A characteristic of humans is that we perform logically distinct and kinematically unrelated actions in parallel. For example, consider "a hand wave sequence." A performer can wave his hands when in different postures — while standing, sitting, talking, and walking. The actions in such cases may be viewed as compositional. The number of such compositional actions tends to grow exponentially. Additionally each action can be performed in multiple styles. For example the hand wave may be performed at different rates. Variations of the same basic motion often depend upon internal factors such as moods and external factors such interaction contexts or physical constraints.

Traditionally, however, data for each actor action is acquired separately. In contrast to the discussion in the previous paragraph, such a scheme has the disadvantage that virtual actor actions are limited to the number of sequences originally captured. This drawback directly impacts interactive applications. In such applications only a few motion sequences corresponding to key actions performed by the virtual actors are used. The reason for this is pragmatic. It is difficult to anticipate and act out "all" the different combinations of actions that would ever be required during runtime. Capturing all the possible correlated combinations is often neither possible nor desired despite the obvious advantages of capturing a large number of combinations.

This Paper

The obvious way to deal with motion composition is to cut and paste motion segments across existing mocap clips. However, such an attempt is likely to result in motions that do not look human as a result of inherent lack of cross-body correlation.

The correlation in a real locomotive sequence may be due to active intentions on part of the actor or as the result of passive reflex. Examples of intentional correlation occur in movements such as relaxed walking, where arms swing out of phase with the legs for energetic reasons. This is a gait that is chosen by the actor, and can be broken at will – for example, to scratch or to reach. Reflex correlations occur as a result of the body reacting to maintain equilibrium. For example, the arms may be extended out in order to balance a fall. If the arm movement in this case is replaced with some other arm movement, the resulting motion may not look human. In either case the correlations play an important role in defining believability of the final motion.

While researchers in the field of behavioral animation have built systems, [Blu95][Per96], that take advantage of parallelism in actions, attempts at automatic composition are recent [Sha04][Ike04]. In this paper we describe our method to synthesize new motions by composing together different actions onto a base clips, taking into consideration the problems discussed above. Our solution is based on a scheme that breaks down the original problem into manageable parts as shown in Figure 1. The net result is (see video) that it generates believable grafts.

To our knowledge motion grafting was first discussed in our own prior unpublished work [Sha04]. An interesting implementation has been subsequently described in [Ike04]. The work presented here complements the work in [Ike04] in the following ways:

- For increased quality, we target only motions that have running, jumping, and walking motions. Several unsuccessful transplants are reported in [Ike04].
- The randomization rules to generate new motion are not used in our work. Instead currently we have used a Cartesian product of upper and lower body classification to generate candidate grafts.
- Instead of using an SVM based classification to determine successful transplants, we use the intrinsic correlation available in cluster graphs.

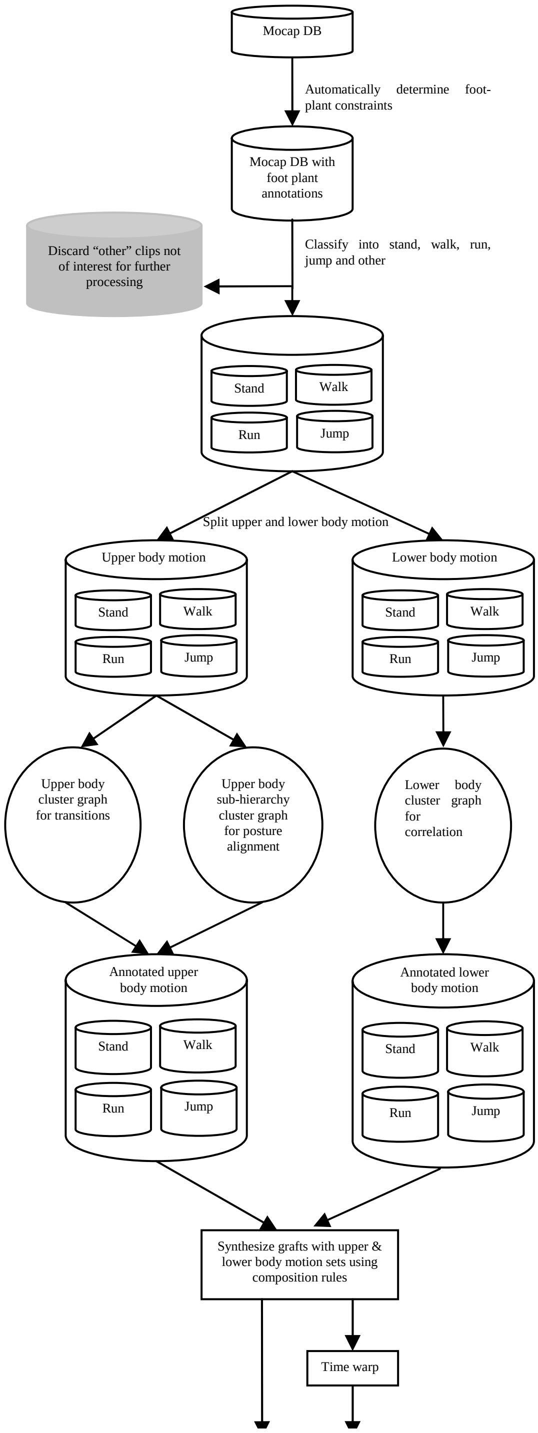

Figure 1 Grafting framework. Our scheme starts with a discovery of locomotive motions from the motion capture database. After classifying independent kinematics chains, a cluster graph data structure is used to correlate seemingly different motions. Correlation is key to generating believable grafts. An optional time warp enables "scaling" in time.

The rest of this paper is organized as follows. We first mention the related work in this area. Next we briefly describe our cluster graph [Bas05] data structure. An overview of our process follows in Section 4. Details of the steps such as determination of independent kinematics chains are described in Section 5. Sample results follow to demonstrate the efficacy of our method (results are best viewed in the accompanying videos).

2. Background

The history of research in humanoid animation dates back more than 15 years. Motion synthesis methods can be broadly classified as kinematics based, dynamics based and constraint based methods. In addition there are also hybrid methods which mix one or more of the other techniques. More recently, methods based on motion captured sequences have been developed.

Behavioral Animation

Behavioral animation researchers have observed and described ways to synthesize parallel actions. [Blu95] describes the use of lock variables to share degrees of freedom between motor skills. [Per96] explicitly allow for action compositing in Improv. Therein they describe a layered architecture for enabling parallel execution of independent actions. The primary difference between these works and ours lies in the fact that they deal with motion sequences explicitly designed to work independently. Defining high fidelity motion sequences is non-trivial. [Per96] construct motion sequences using combinations of sine, cosine and coherent noise. Noise is used to generate variations in motion. Sequences generated in this manner, though not repetitive, do not contain individualistic nuances of the performer. We choose to work with motion captured data which is richer. However the independent actions are not well defined here.

Gait Synthesis Techniques

[Mul02] provide an excellent survey of computer animation of human walking. [Sun01] describe a low-level gait generator based on sagittal elevation angles, which allows curved locomotion to be created easily. They also describe an inverse motion mapping algorithm that allows motion to be adapted to uneven terrains. In addition they describe a higher level control frame work that allows motion requirements to be specified at a high level by sketching the desired path. [Hod95] describe an algorithm that allows simulation of running, bicycling and vaulting. The simulation is achieved through control algorithms that cause physically realistic models to perform the desired behavior. [Fal01] describe a method to combine various physically based simulation controllers into a unified framework.

Mocap based Motion Editing Techniques

Mocap based techniques, unlike synthesis techniques described above, start with an existing motion and adapt it to different requirements. Researchers working in this field have proposed a number of innovative techniques adapting signal processing methods or employing constraint based solvers. [Bru95] have successfully applied techniques from image and signal processing domains to designing, modifying and adapting animated motions. [Unu95] describe a method for modeling human figure locomotion's with emotions. Herein Fourier expansions of experimental data of actual human behaviors serve as a basis from which to interpolate or extrapolate the human locomotion's. [Wit95] describe a simple technique for editing captured animation based on warping of the motion parameter curves. [Gle97][Gle98] use space-time constraint formulations to modify captured motion and retargeting motion to new characters with different segment lengths. [Gle01] provides a comparison of constraint based motion editing methods. [Lee99] describe a hierarchical framework for adapting existing motion of humanoids to externally specified constraints. [Shi03] describe a method that takes into consideration physical principles to touch up synthetically generated motion, so as to make it more plausible.

Mocap Based Synthesis Techniques

Motion synthesis techniques create new motion from existing motion data. [Kov02a], [Lee02], [Ari02] describe techniques to create new motion sequences from a corpus of motion data. Each technique essentially clusters similar motion into nodes. The next phase builds a graph of nodes, where each edge represents a transition between nodes. A walk through the cluster node graph results in synthesis of new motion sequences. The techniques differ in metrics used for clustering, pruning schemes and control criteria for node walk. [Pul00][Pul02] describe a scheme for synthesizing missing degrees of freedom and adding details to specified degrees of freedom, for a roughly specified motion sequence. Their method uses the various correlation between the various degrees of freedom (DOF's) within each motion sequences. Forsyth et. al [2] describe a technique using a novel search method based around dynamic programming to interactively synthesize motion from annotations. Here the system synthesizes motion corresponding to an annotated timeline painted by the user.

[Gle03] present a technique that preprocesses a corpus of motion capture examples into a set of short clips that can be concatenated to make continuous streams of motion. The resulting simple graph structure can be used in virtual environments where control and responsiveness are more important than accuracy. [Bra00] use machine learning techniques to model motion sequences as stylistic HMM parameterized by a style vector. Motion can be synthesized in a number of ways - a random walk over the HMM states, by trying to match given input sequence to an optimum set of HMM states etc. However, as the method is statistical there is no direct control over the desired motion.

3. CLUSTER GRAPH

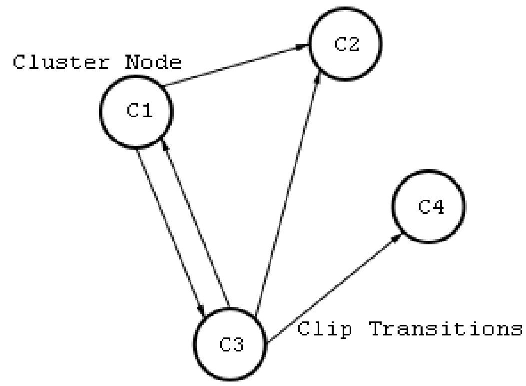

Figure 2 The cluster graph data structure

We describe the cluster graph [Bas05] in brief as our grafting technique uses cluster graphs extensively. The cluster graph is a versatile data structure that we use to cluster together frames from different clips based on similarity. A cluster graph

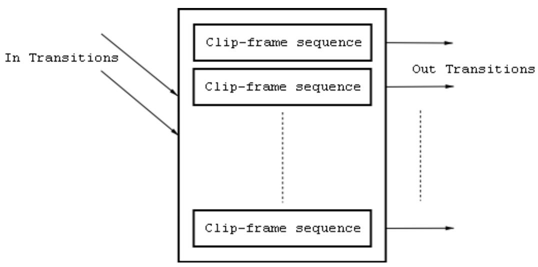

Figure 3 A cluster graph node

automatically chops individual clips and collects similar sub-clip sequences together. Each subsequence could be even half a dozen frames long. Seemingly unrelated clips are brought together into these nodes increasing the choices available for re-synthesis.

Nodes in a cluster graph contain frames from one or more clips. Frames within a node are "similar," that is, the error between any two frames is below a threshold. Edges are obtained from the natural sequential ordering of clips within nodes. Figure 2 shows an example.

Within a node (Figure 3), the frames are sorted by clips and time. Contiguous sequences of frames are collected together into a structure called clip-frame sequence. We maintain out-transitions for each clip-frame sequence for each cluster node. Maintaining one transition per clip-frame sequence automatically prunes away transitions from contiguous frames.

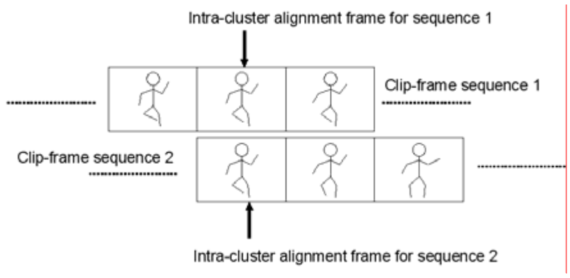

Figure 4 Intra-cluster alignment frames

Once clip-frame sequences are clustered in a graph, each cluster node contains similar frames. We then find an intra-cluster alignment frame, (Figure 4) for each clip in the cluster, using a correlation procedure. This alignment frame, is the best point of transition between any two clip-frame sequences in the cluster.

Cluster graph advantages

Cluster graph nodes contain clip-frame sequences from different mocap sequences clustered together based on similarity. These clustered clip-frame sequences establish correlation amongst clips.

Cluster graphs provide a level of granularity smaller than those of motion graphs [14]. Traditional motion graphs record only one transition point between each pair of clips. As a result this data structure has been used to find transitions between two clips without enforcing a hard time constraint. For the real-time version, time is an important factor. We may need to transition between two clips at precise, or more controlled instants in time. Therefore it is useful to maintain as many distinct transition points as possible. Note further that transition points occur in bunches. Cluster graphs prune multiple transition points lying very close to each other temporally.

4. METHOD OVERVIEW

Our aim is to allow motion re-synthesis by selectively grafting motion signals captured on different actions on to a base signal. In order to do this we use a collection of techniques and steps as illustrated by Figure 1.

1. We start with a database of motion captured clips.
2. Every clip in the mocap database is preprocessed to detect foot plant constraints. We annotate the clips with this information.
3. We classify the clips based on the foot plant annotation into the following categories – "stand", "walk", "run", "jump" and "others." The clips of interest to us are the ones labeled "stand", "walk", "run" or "jump". We discard the clips labeled "other".
4. We separate the upper body and lower body motion signals based on their respective *independent kinematics chains*. We retain the respective upper and lower body pair correspondence during this process for later use.
5. We create a cluster graph for lower body motion. We use this to obtain correlation information amongst lower body motion. Lower body motion forms the base clip upon which grafting occurs.
6. We create two cluster graphs for the upper body motion sets. The first is created with the entire upper body and is used for grafting transition information. However, only a subset of transitions will not violate self penetration. A second cluster graph is created with just the *root – lowerback – upperback – torso* joint chain, to conservatively estimate safe grafts.
7. We synthesize grafts as a Cartesian product of upper body and lower body motion sets. We use the composition rules explained below.
8. As an optional last step we allow the animator to accept or reject generated clips before enhancing the motion database.

4.1.1 Composition Rules
Our categorization of motion results in upper and lower body motion divided into four sets corresponding to "stand", "walk", "run" and "jump". We synthesize graft motion grafts by taking certain conservative Cartesian products of upper and lower body motion sets:

1. (Upper body motion set "stand") X (Lower body action sets "stand", "walk", "run" and "jump").
2. (Upper body motion set "walk") X (Lower body motion sets "walk" and "run")
3. (Upper body motion set "run") X (Lower body motion sets "walk" and "run")
4. (Upper body motion set "jump") X (Lower body motion set "jump")

5. GRAFTING
Grafting is the process of synthesizing motion. It essentially involves masking out the original base clip signal for the selected set of DOF's and replacing them with those from a different clip. In this section we give some details of the modules in section 4.

Independent Kinematic Chains
We observe that a skeletal hierarchy essentially contains several kinematics chains as in Figure 5. For example nodes [root, lhipjoint, lfemur, ltibia, lfoot, ltoes] constitutes a kinematics chain. For different chains having a common root, the kinematics inter-dependency is restricted to the

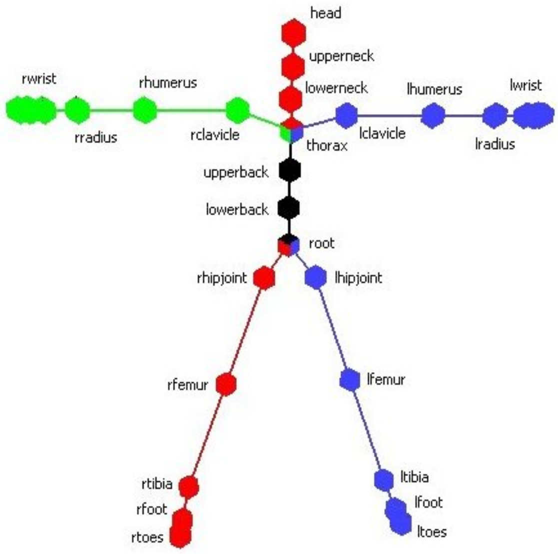

Figure 5 Hierarchical skeleton with independent kinematics chains demarcated

common roots DOF variables. The chains are unaffected by DOF variables at non root nodes. We call such chains "independent chains." The motion signals captured for each independent chain constitutes an "independent action." We think of a motion captured sequence to be composed of several "independent actions" running in parallel. We graft motion on to DOF's of these independent kinematics chains. Cross body correlation creates inter-dependence amongst independent actions. We use independent kinematics chains to maintain correlation. We also use these kinematics chains to create sub-hierarchy cluster graphs.

Six such kinematics chains are defined in Figure 5 (seen best in color).

Chain 1: Thorax … Head

Chain 2: Thorax … LWrist

Chain 3: Thorax … RWrist

Chain 4: Root … Thorax

Chain 5: Root … LToes

Chain 6: Root … RToes

Clip classification

We use the lower body kinematics chains for clip classification. One of the important characteristics of lower body motion is the foot-plant constraint. We use foot-plant constraint pattern matching to classify clips. We make the following observation regarding foot-plant constraints.

1. Standing Stationary: Both feet remain planted.
2. Walking: Alternate feet are planted passing through a double step pose.
3. Running: Alternate feet are planted with intervening stages of both feet being of the ground.
4. Jumping: Both feet are either planted or in air simultaneously.

We encode the foot-plant behavior as a set of string symbols and use simple string matching to classify clips.

5.1.1 Detecting foot plant constraints

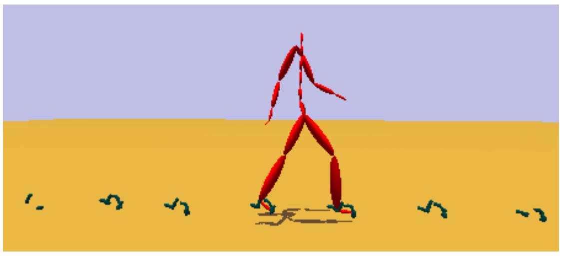

Figure 6 Frames with heel or ball joints stationary detected as foot-plant frames

We identify foot plant constraints by first identifying frames with zero crossings for vertical displacement (the y axis in our case). We select all frames which are close to the ground, within a given threshold. This forms the seed set of foot plant frames. For most normal walk sequences, we observe that the foot is placed on the ground for more than one frame. However, in a motion captured sequence, the foot positions may not coincide exactly due to foot skate. [Kov02b] describe a technique to identify and correct foot skate. We use a simpler method. From the initial set of foot plant frames obtained above, we sequentially search in both directions and cluster frames, near the seed foot plant frame, where the magnitude of the displacement vector is below a given threshold value. We stop the search at the first frame that fails the test. We then cluster together, like foot plant frames based on their sequence in the clip.

6. RESULTS

Our motion capture database, after categorization, consists of more than a 100 clips from the CMU motion capture database. As noted earlier the cluster graph data structure chops individual clips into smaller clip-frame sequences each of which can be as small as half a dozen frames. This combined with the ability of the cluster graph to detect and encode loops, results in an order or two magnitude increase

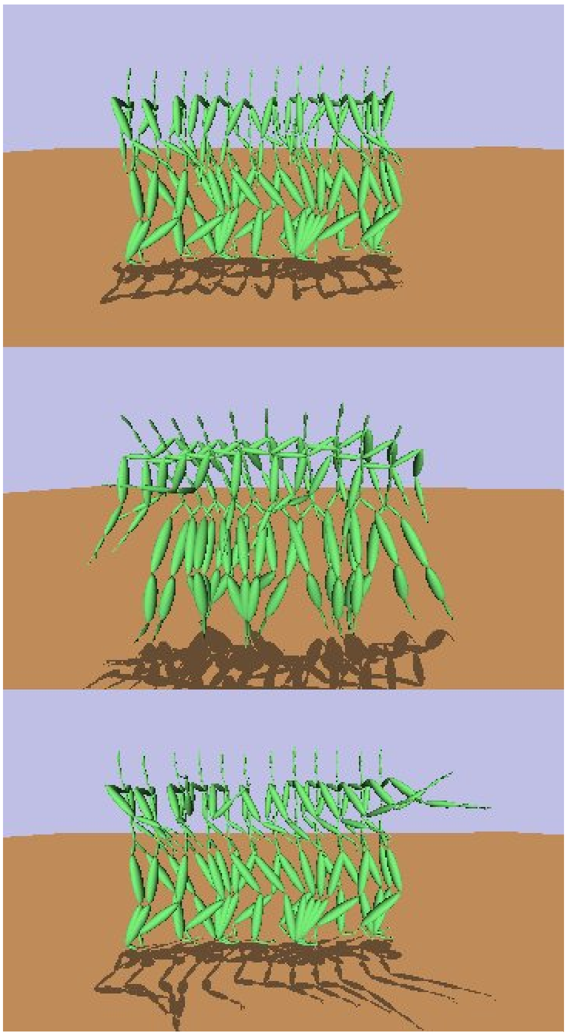

Figure 7 The walk clip (top) is used as the base clip onto which the basket ball dribble (middle) is grafted to synthesize new (bottom) clip.

ISBN 80-86943-03-8

in the large number of paths through the structure.

Figure 7 and Figure 8 are representative of the results obtained using our method. In Figure 7 The walk clip (top) is used as the base clip onto which the basket ball dribble (middle) is grafted to synthesize new (bottom) clip. In Figure 8 The normal walk clip (top) is used as the base clip onto which arm motions from the exaggerated stride (middle) is grafted to yield the marching like motion (bottom). As can be seen (from the accompanying videos) the results are fairly believable and smooth.

7. CONCLUSION

In this paper we have described our method to enhance a collection of motion captured clips by synthesizing new motion. Our synthesis technique composes together motion onto independent kinematics chains of a base clips. Our composition rules allow us to take into account the correlation between different actions. We use cluster graphs extensively to obtain correlation information.

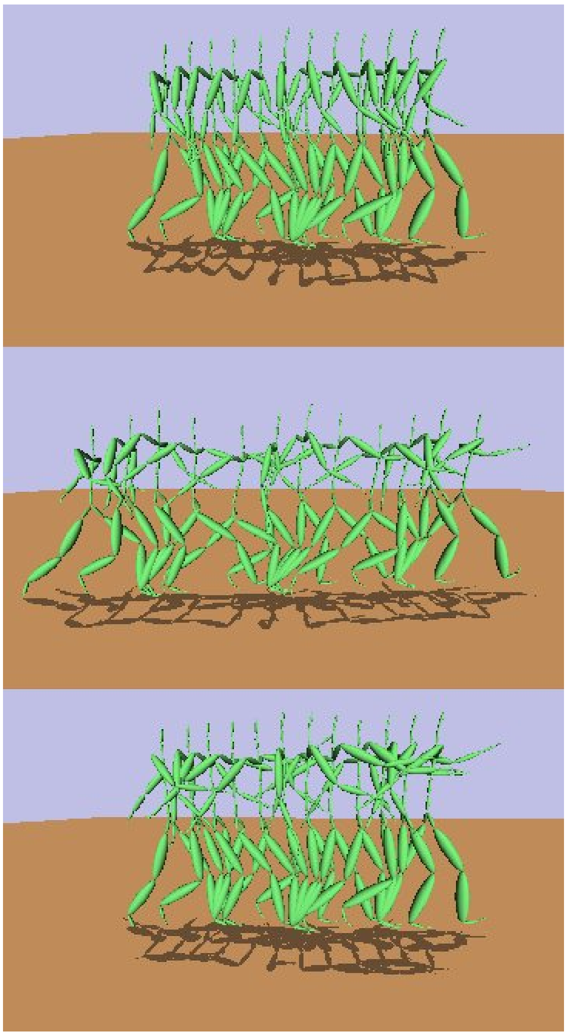

Figure 8 The normal walk clip (top) is used as the base clip onto which arm motions from the exaggerated stride (middle) is grafted to yield the marching like motion (bottom)

8. REFERENCES

[Ari02] Arikan, O. Arikan and Forsyth, D. A. Interactive Motion Generation from Examples. In Proc. of Siggraph '02, pp. 483 - 490, 2002.

[Ari03] Arikan, O., Forsyth, D. A., and O'Brien, J. F. Motion Synthesis from Annotations. In Proc. of Siggraph '03, pp. 483 - 490, 2003.

[Bas05] Basu S. K., Shanbhag, S., and Chandran S., Search and Transitioning for Motion Catpured Sequences. ACM Symposium on Virtual Reality Software and Technology 2005.

[Blu95] Blumberg, B.M and Galyean, T. A., Multilevel Direction of Autonomous Creatures for Real-Time Virtual Environments, Proceedings of Siggraph '95, 1995, pp. 47-54.

[Bra00] Brand, M. and Hertzmann, A. Style machines. In Proceedings of Siggraph '00, 2000.

[Bru95] Bruderlin, A. and Williams, L. Motion Signal Processing. In Proc. of Siggraph '95, 1995.

[Fal01] Faloustos, P. , van de Panne, M., and Terzopoulos, D. Composable controllers for physics-based character animation. In Proceedings of Siggraph '01, August 2001.

[Gle97]Gleicher, M. Motion Editing with Spacetime Constraints.In Proc. of the 1997 Symposium on Interactive 3D Graphics.

[Gle98] Gleicher, M. Retargetting Motion to New Characters. In Proc. of Siggraph '98, 1998.

[Gle01] Gleicher, M. Comparing Constraint-based Motion Editing Methods. Graphical Model, 63(2):107 - 134, 2001.

[Gle03] Gleicher, M, Shin, H. J., Kovar, L., and Jespen, A. Snap together motion: Assembling run-time animation. In 2003 Symposium on Interactive 3D Graphics, April 2003.

[Hod95] Hodgins, J. K., Wooten, W. L., Brogan, D. C. , and O'Brien, J. F. Animating human athletics. In Proceedings of Siggraph '95, 1995.

[Ike04] Ikemoto, L., and Forsyth, D., A., Enriching a Motion Collection by Transplanting Limbs, Proc. ACM Symposium on Computer Animation, 2004.

[Inm89] Inman, V. T. , Ralston, H. , and Todd, F. Human Walking. Lippincott Williams & Wilkins, 1989.

[Kov02a] Kovar, L., Gleicher, M., and Pighin, F. Motion Grahs. In Proc. of Siggraph '02, 2002.

[Kov02b] Kovar, L., Schreiner, J., and Gleicher, M. Footskate cleanup for motion capture editing. In Proceedings of the 2002 ACM Symposium on Computer Animation (SCA), July 2002.

[Lee99] Lee, J. and Shin, S. Y. A Hierarchial Approach to Interactive Motion Editing for Human like Figures. In Proc. of Siggraph '99, 1999.

[Lee02] Lee, J., Chai, J., Reitsma, P. S. A., Hodgins, J. K., and Pollard, N. S. Interactive Control of Avatars Animated with Human Motion Data. In Proc. of Siggraph '02, 2002.

[Mul02] Multon, F., France, L., Cani-Gascuel, M.-P., and Debunne, G. Computer animaton of human walking: A survey. Technical Report 3441, INRIA, June 1988.

[Per96] Perlin, K and Goldberg, A., Improv: A System for scripting interactive actors in virtual world, Proceedings of Siggraph '96, 1996, pp. 205-216.

[Pul00] Pullen, K. and Bregler, C. Animating by Multi-level Sampling. In Proc. of IEEE Computer Animation 2000, 2000.

[Pul02] Pullen, K. and Bregler, C. Motion Capture Assisted Animation: Texturing and Synthesis. In Proc. of Siggraph '02, 2002.

[Shi03] Shin, H. J., Kovar, L., and Gleicher, M. Physical touch-up of human motion. In Pacific Graphics 2003, October 2003.

[Sha04] Shanbhag, S., and Chandran, S., Parallel Action Synthesis for Skeletally Animated Characters, Technical Report IIT Bombay, February 2004.

[Sun01] Sun, H. C. and Metaxas, D. N. Automating gait generation. In Proceedings of Siggraph '01, August 2001.

[Unu95] Unuma M., Anjyo, K., and Takeuchi, R. Fourier principles for emotion based human figure animation. In Proceedings of Siggraph '95, 1995.

[Wit95] Witkin, A. and Popovic, Z. Motion Warping. In Proc. Of Siggraph '95, 1995.